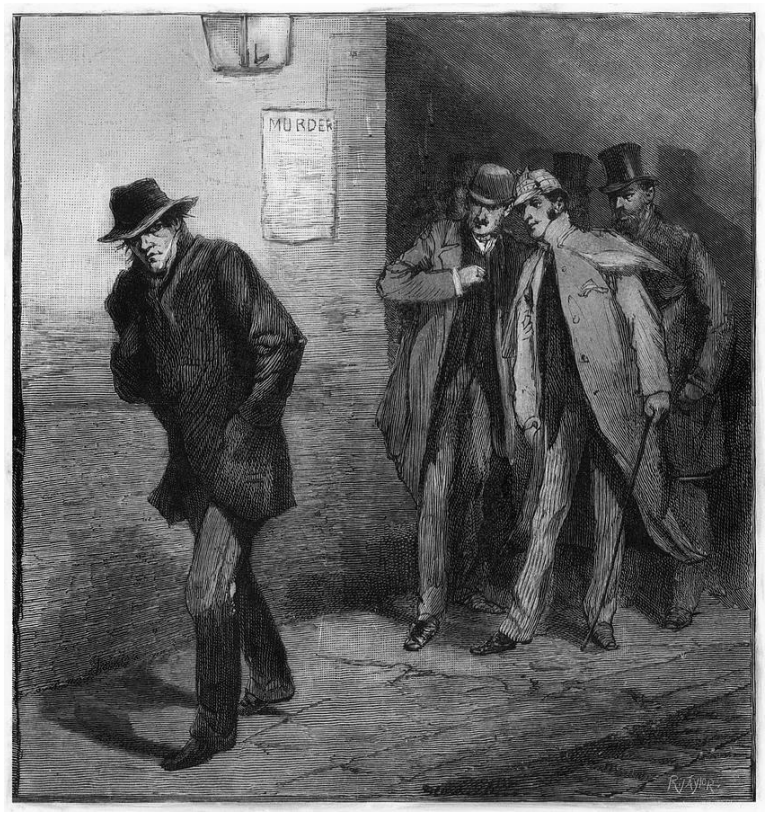
Creating Traceability in Your DevOps Work

Chris B. Behrens
SOFTWARE ARCHITECT
@chrisbbehrens

The Power of Traceability



Jack the Ripper

Different people suspected of the murders in Whitechapel

It was probably someone with no paper trail

A truly untraceable life is nearly impossible today

Even the Unabomber had an address

And that was AFTER he went to Harvard

My Point

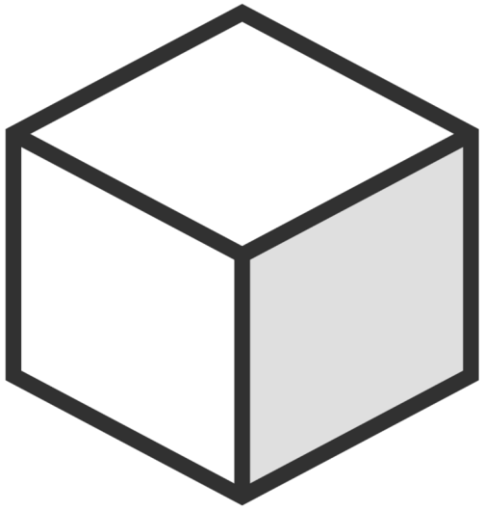
We must create traceability

And we need to do it ahead of time

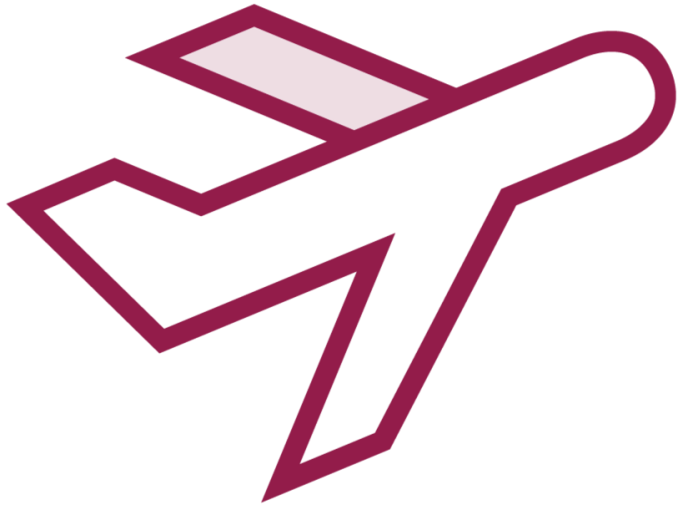
Or else it's a lot more expensive and time-consuming

If we had better documentation, we might know who Jack the Ripper was

Traceability and Root Cause Analysis



The “black box”, a flight data recorder



Plot twist – the black box is never black, they’re orange

Flight Data Recorders record many parameters of data

They usually can tell you that a component failed

Then you can cross-reference maintenance records for the component

Who maintained it?

What did they do?

Where they properly trained?

Why was it possible for negligence to occur in the first place?

Why You Should Make Your Peace with Build Failures

Failed builds are your friends

Better for your build to find the problem than a customer

Root Cause Analysis in Builds

The proximal cause – the cause near to the failure

The ultimate case – the root cause of the failure

The Challenger Shuttle disaster

Feynman demonstrated the proximal cause

- An o-ring which sealed the oxidizer tanks
- An o-ring that was too cold to function properly

"...management, on the other hand, claims to believe the probability of failure is a thousand times less...demonstrating an almost incredible lack of communication between themselves and their working engineers."

Dr. Richard P. Feynman, https://bit.ly/37K3UpG

Traceability and Your Role as a Build Engineer

The proximal reason is usually right there in the logs

The ultimate reason is more important

A bad story to begin with

This should have been downmerged first

Teach Aziz how to downmerge

Only the build engineer has this view of things

Avoid scheduled builds

The Nuts and Bolts of Traceability in Azure DevOps

Scheduled builds may have made sense once

But not with a proper Git workflow

The build should be a *test* of that increased confidence

If they really believed in it, they'd create a pull request

What You Do Instead

Builds should be triggered from a merge, or other commits

“This happened to test Derek’s assertion that this code is ready”

The Next Step

With a commit-triggered build…

- We need to understand why the commit happened
- Ideally, as a user story

When you establish this link

- You get traceability
- The value of creating the story well becomes obvious
- And groomed with failure in mind
- With greater detail, or else
- You'll have to explain why when everything fails

Azure DevOps and GitHub

Integrating DevOps Builds with GitHub

Build integration is job #1

GitHub operations trigger a build

And the build runs normally

How This Is Done

You establish a connection between Azure DevOps and GitHub

The connection enables traceability

You simply enter “AB#17”, for “Azure Boards”

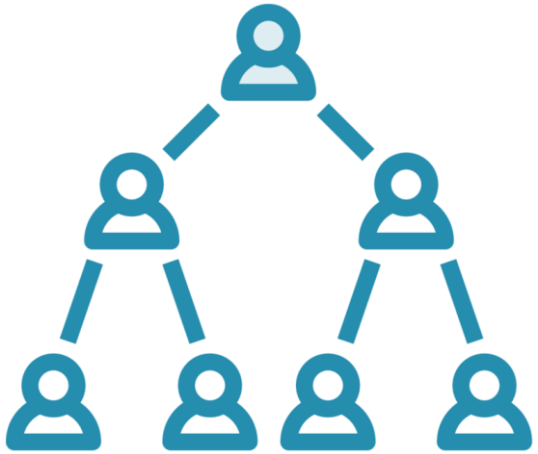
Covered previously in Designing Build Automation

Integrating Azure Boards with GitHub

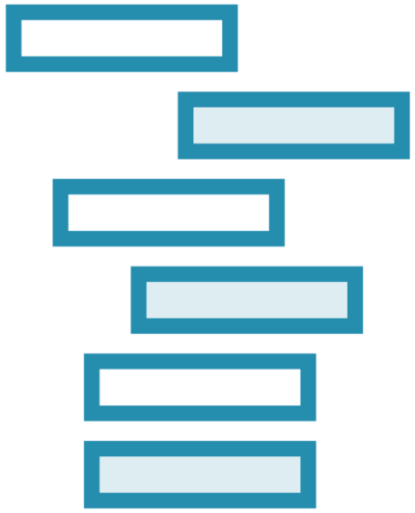
I have to learn new stuff a lot

But you should give your team time to adjust

The Best of Both Worlds



Company mergers and acquisitions



An integration between Boards and GitHub

PRs percolate automatically to GitHub

Validated Learning

A topic worth weighing in on

My actual academic field, Economics

What Is It?

A process

A way of asking questions

In an efficient way

Friedrich Hayek and “The Pretence of Knowledge”

- The examination of premises
  - What our premise is
  - That the premise exists
- Nobel prize for Economics in 1974
- https://www.nobelprize.org/prizes/economic-sciences/1974/hayek/lecture

“If man is not to do more harm than good in his efforts...he will have to learn that in this, as in all other fields where essential complexity of an organized kind prevails, he cannot acquire the full knowledge which would make mastery of the events possible.”

Friedrich A. Hayek

Poison Premises

When we act from false premises...

At best, we create waste

At worst, we create harm

Is it a DDOS attack?

Or just a misconfigured firewall?

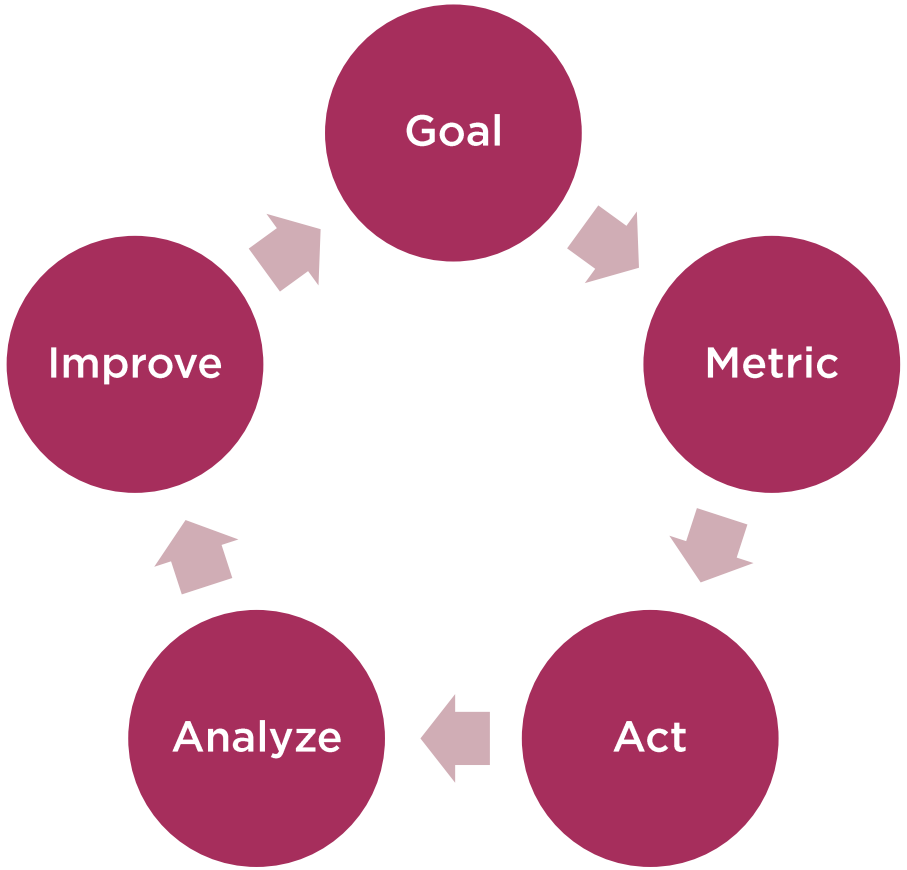
What are different explanations which account for the evidence?

How Validated Learning “Fixes” This

1. Bandwidth being consumed by a shared resource
2. An update to the firewall has degraded performance
3. An attacker is performing a denial-of-service attack

Validate assumptions FAST

The Validated Learning Cycle



A week is long enough to determine whether the UI change affects sign-up rate.

A UI change alone can significantly affect sign-up rate.

“The first principle is that you must not fool yourself — and you are the easiest person to fool.”

- Richard P. Feynman

Course Summary

Dashboards in Azure

Creating policies to control costs

User Analytics

Traceability

- Root Cause Analysis
- Implementing traceability across your Azure process

Integrating Pull Requests in GitHub with Azure

Validated Learning in Agile

- The Pretence of Knowledge
- A process to validate the knowledge we believe we have

THANK YOU FOR WATCHING!!!